Recurrent Neural Networks in deep machine learning to predict patient mortality

**Introduction**

This research demonstrates the application of deep learning models in practical situations. Usually, we are faced with a larger amount of medical record data without knowing how to classify the statistics, but with this study it is possible to analyze the huge medical database and discover the patterns of disease emergence. Building a deep neural network that can predict future outcomes based on time series, or sequential, data. Helping people solve challenging problems using AI and deep learning.

In addition, personalized predictive medicine necessitates the modeling of patient illness and care processes, which inherently have long-term temporal dependencies. Healthcare observations, stored in electronic medical records are episodic and irregular in time. Deep neural network models have conventionally been designed, and experiments were performed upon them by human experts in a continuing trial-and-error method. This process demands enormous time, know-how, and resources. To overcome this problem, a novel but simple RNN model is introduced to automatically perform optimal predictive tasks with deep neural network architecture.

**Methodology**

Model establishment and data processing need to be written in Python, including keras API, numpy, pandas and matplotib packages. Keras API runs on GPUs and CPUs and allows for easy and fast prototyping. Pandas is an open-source, it used in academia and commercial domains. It is deal with efficient dataframe object for data manipulation with integrated indexing. NumPy is a Python scientific computing package, support for large, N-dimensional arrays and matrices. It’s a collection of high-level mathematical functions and a basic part for building modular neural network. Matplotlib is a Python 2D plotting library producing publication quality figures. Using it to process data for visualization can be clearer and more intuitive, and is an important step in the display of deep language learning results.

**Modeling**

A [Recurrent Neural Network](https://www.sciencedirect.com/topics/computer-science/recurrent-neural-network) (RNN) is a [neural network](https://www.sciencedirect.com/topics/computer-science/neural-networks) repeated over time. In particular, an RNN allows self-loop connections and shared parameters across different time steps. While a [feedforward neural network](https://www.sciencedirect.com/topics/computer-science/feedforward-neural-network" \o "Learn more about Feedforward Neural Network from ScienceDirect's AI-generated Topic Pages) maps an input vector into an output vector, an RNN maps a sequence into a sequence (Figure1.)



Figure1. RNN illustration

Long Short-Term Memory (LSTM) is a RNN that effectively solves the vanishing gradient problem. Central to an LSTM is a linear self-loop memory cell that allows gradients to flow through long sequences. The memory cell is gated to moderate the information flow to or from the cell (Figure2.). LSTMs have been successful in many applications, such as machine translation, handwriting recognition and speech recognition. In our network, the sequence of measurements recorded from an encounter will be used as input to the network, and a probability of survival prediction will be generated. It enables real-time monitoring of the patient’s probability of survival and insight into the patient’s trajectory.

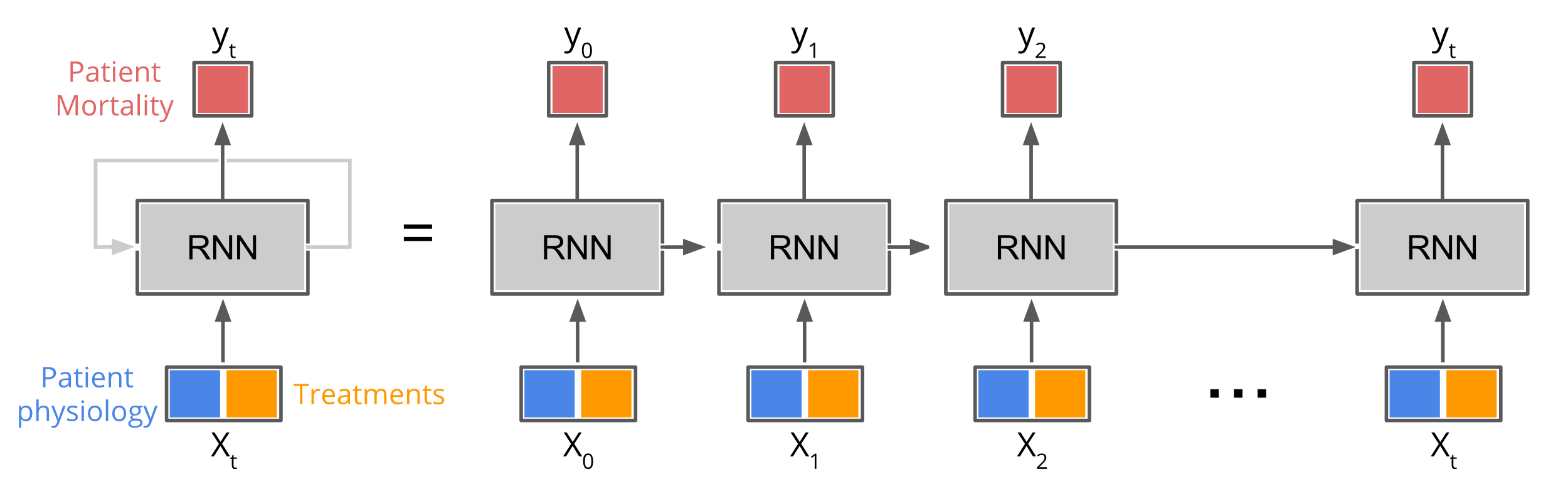
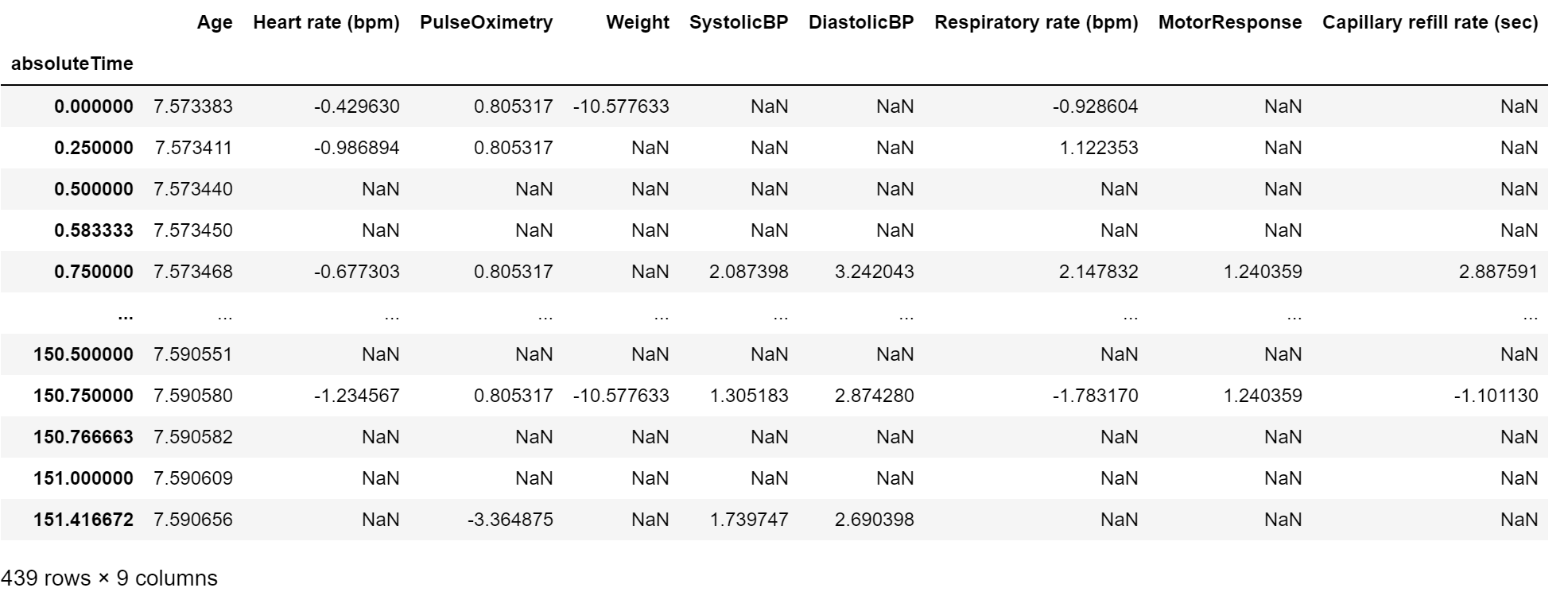


Figure2. Long Short-Term Memory illustration

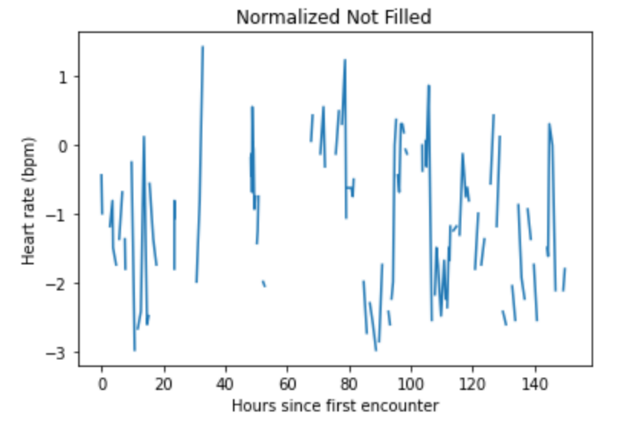
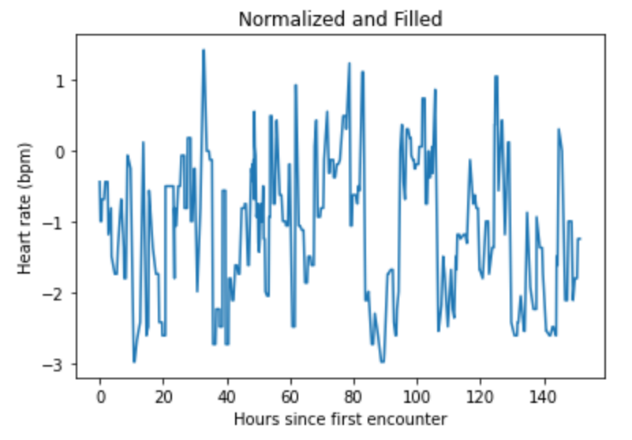
**Discussion**

An electronic medical record (EMR) is a digital version of a patient’s health information. Data provided by PICU at Children’s Hospital Los Angeles, it contains 5,000 unique patient encounters in the training set, each encounter includes multiple observations during the hospital stay, each observation includes values in some of the 265 measurement variable categories. Data is an irregular time series of measurements taken over the course of a patient’s stay in the PICU. Not all measurements were taken for all patients, thus the first priority is to clean up the data and set dependent variable. I set alive as 1, not alive as 0. There are 1,113,529 observations (rows) containing 265 variables (columns) over the 5000 patients in the training data set.

First, the data is processed in three stages. Normalize data, fill data gaps, pad and truncate the data sequences using Pandas and Numpy. With a lot of missing data, I picked out target data to fill in and normalized data (Figure3&Figure4.).



(Figure3. Data Normalization)

****

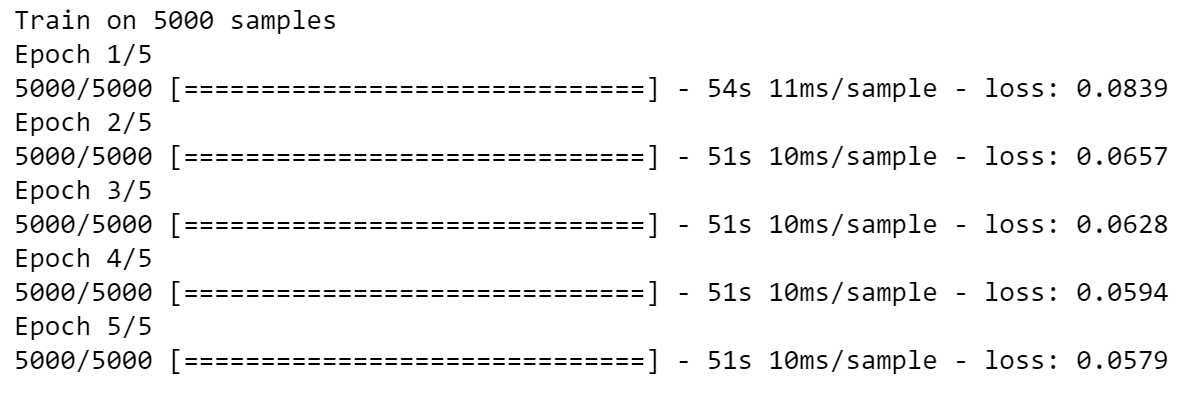
(Figure4a.Normalized Heart rate with hours) (Figure4b.Filled data)

(Figure4.Normalized and Filled Heart rate with hours)

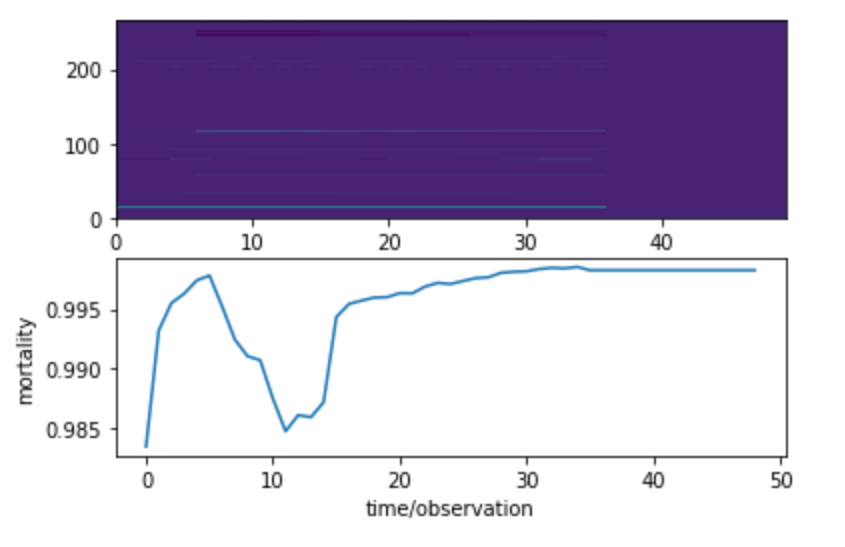
Afterwards, creating the model. Building a Time Series Data Model with Keras and training model. Evaluating the model using validation data and visualize results.

**Conclusion**

Now, feeding some data into the network for training. Our objective is to train the model on past sequential known data so that we can use the model as a predictor on new data. I use a batch size of 128 which means that we update parameters every 128 time steps. For prevent overfitting I will use only 5 training epochs, which means that run through the entire data set 5 times. As can be seen in Figure 5, the loss decreases as the training number increases.

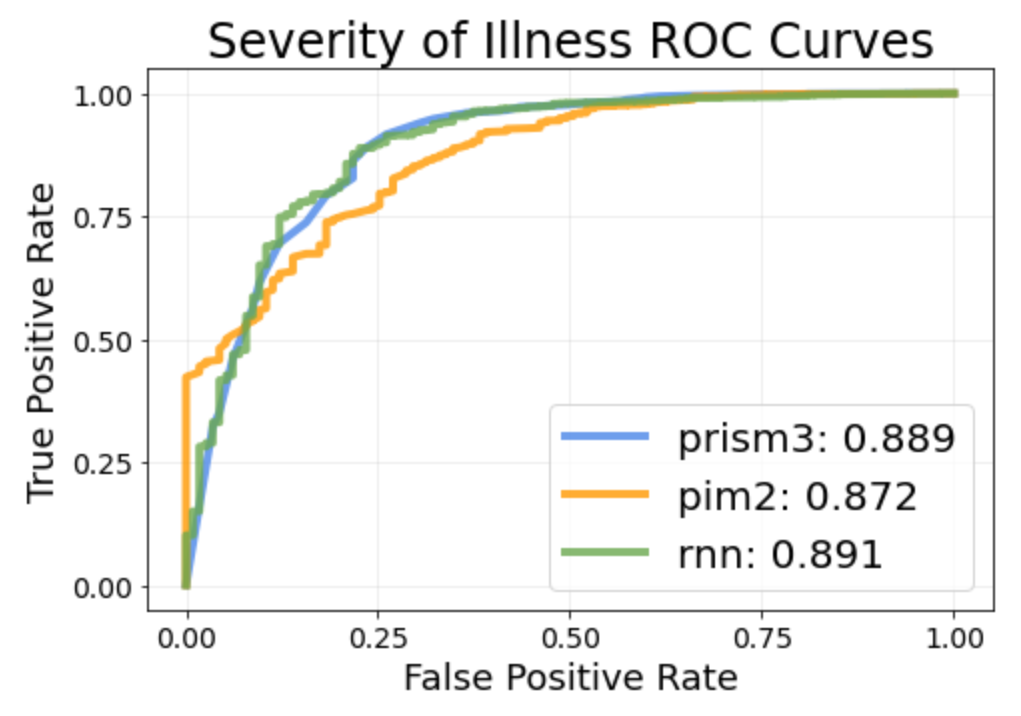
(Figure5. Model training result)

Evaluating the model performance is to predict mortality. The validation set is smaller than the 5000 encounters we used for the training set. We have 2,690 patient encounters for testing. We can see the visualize result from Figure 6, I plot the observation chart for the random patient encounter. The lowest mortality rate was observed at time 10. As the time increases, the mortality rate gets higher.



(Figure6. the patient survivability prediction)

Finally, a comparison is made with the professional medical standard line to see the performance of this model prediction (Figure7.). Compare against baselines: PRISM3 and PIM2. Both PIM2 and PRISM3 are scoring systems for ICU and surgical patients. Models that predict the risk of death of groups of patients admitted to intensive care are available for adult, pediatric and neonatal intensive care. By adjusting for differences in severity of illness and diagnosis, these models can be used to compare the standard of care between units and within units over time. They can also be used to compare different methods of organising intensive care. Estimating mortality risk is also an important component of comparing groups of patients in research trials.



(Figure7.Comparation with standard line)

RNNs provide a method to quickly extract clinically significant information and insights from available EHR data. The amount of data, model complexity, number of features, and number of epochs have been reduced in this research to reduce computational burden. The research displays the performance of a fully trained RNN on a larger dataset. They also show the performance of PIM2 and PRISM3, two standard scoring systems, as well as the performance of a logistic regression model and a multi-layer perceptron (MLP). In time series problems like our prediction project, we need a special type of neural network that includes past information as part of its input. The temporally dynamic nature of the RNN enables it to extract more information from the underlying EHR than an MLP. Overall, LSTM network shows excellent memory function with prediction.